EP 24618 (2)

(12) UK Patent Application (19) GB (11) 2 337 347 (13) A

(43) Date of A Publication 17.11.1999

(21) Application No 9903612.1

(22) Date of Filing 18.02.1999

(30) Priority Data
(31) 9804111 (32) 27.02.1998 (33) GB

(51) INT CL[6]
G06F 9/46

(52) UK CL (Edition Q)
G4A AFT

(56) Documents Cited
EP 0438021 A2

(58) Field of Search
UK CL (Edition Q) G4A AFT
INT CL[6] G06F 1/04 1/06 1/12 9/46
ONLINE: EPODOC JAPIO WPI

(71) Applicant(s)
Lucas Industries public limited company
(Incorporated in the United Kingdom)
46 Park Street, LONDON, W1Y 4DJ, United Kingdom

(72) Inventor(s)
Francis Beresford Dickens

(74) Agent and/or Address for Service
Marks & Clerk
Alpha Tower, Suffolk Street Queensway,
BIRMINGHAM, B1 1TT, United Kingdom

(54) Abstract Title
Distributed data processing system

(57) A distributed data processing system for use in aircraft engine control comprises a plurality of independent cyclically operating processor units (10, 11, 12). One of these is a fuel control (12) acting as a synchronisation master. The others include fan-case (10) and core (11) units, for receiving engine operating parameters, and are interconnected with the master by a communication system. The master periodically transmits synchronisation messages to the other units. The time frame of each unit is shifted depending on a comparison between the timing of these messages and its own independent time frame, based on counting ticks.

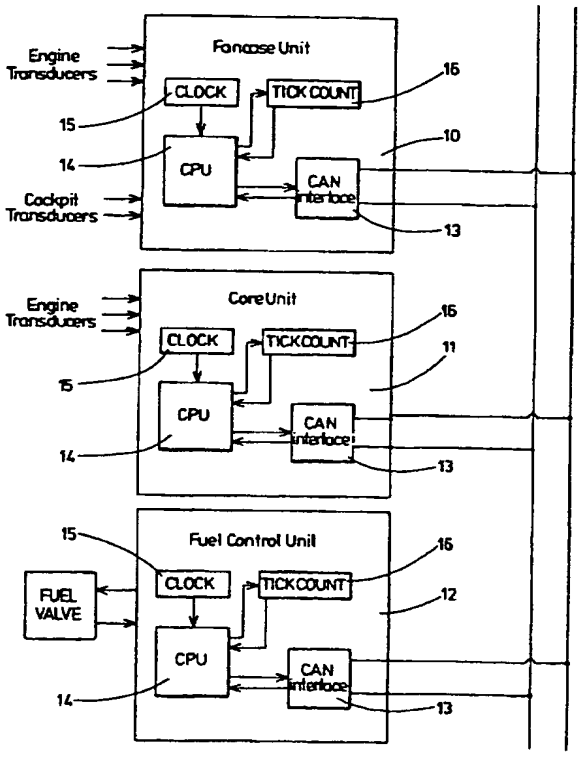

FIG 1

GB 2 337 347 A

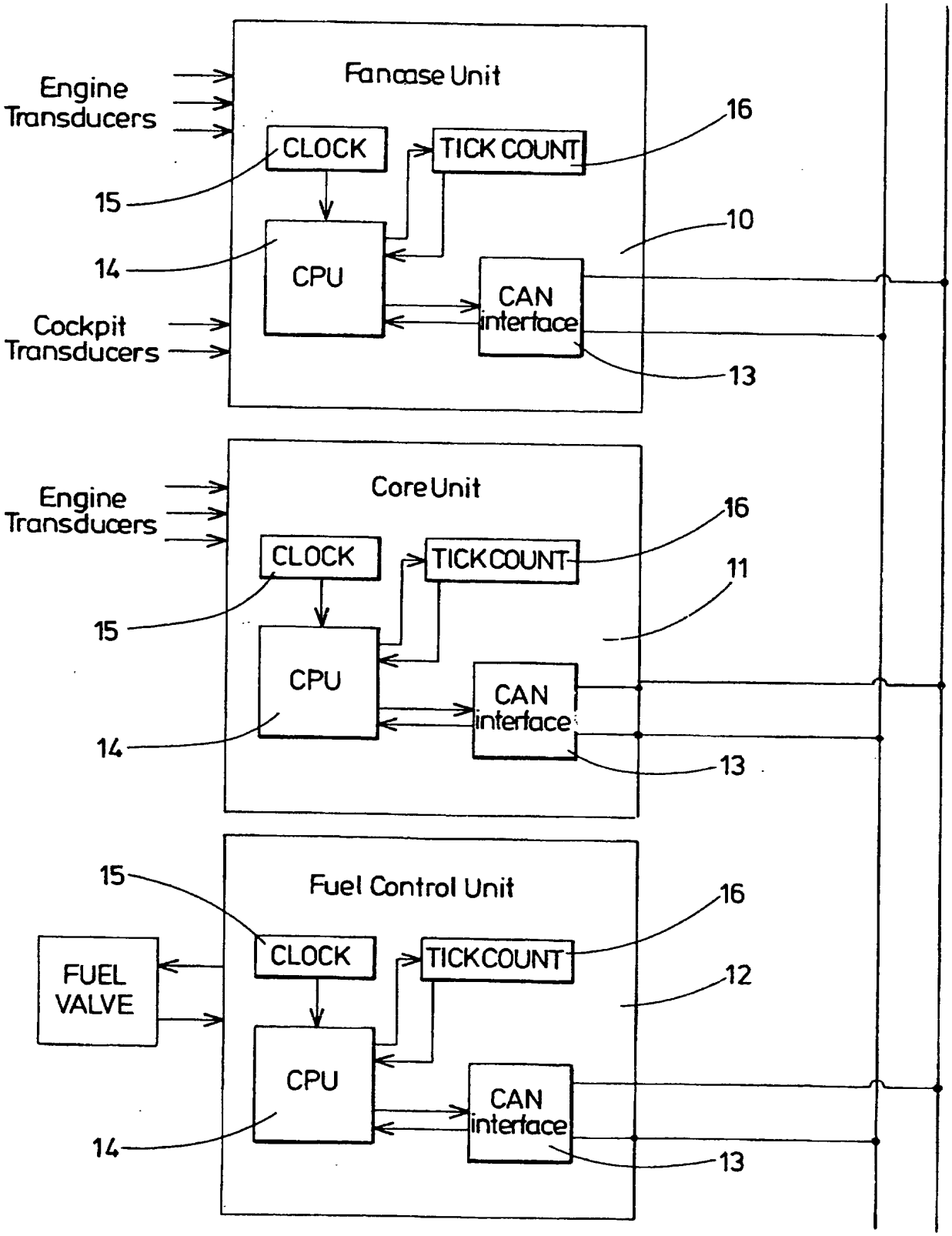

FIG 1

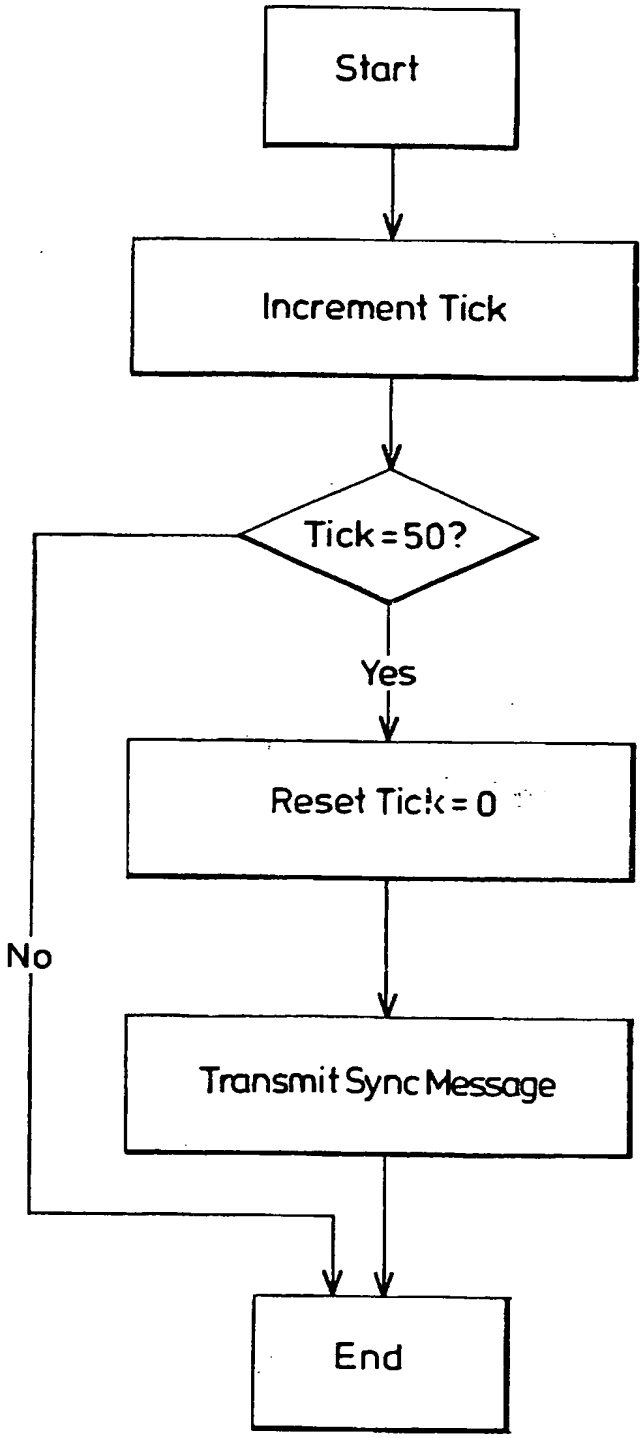

FIG 2

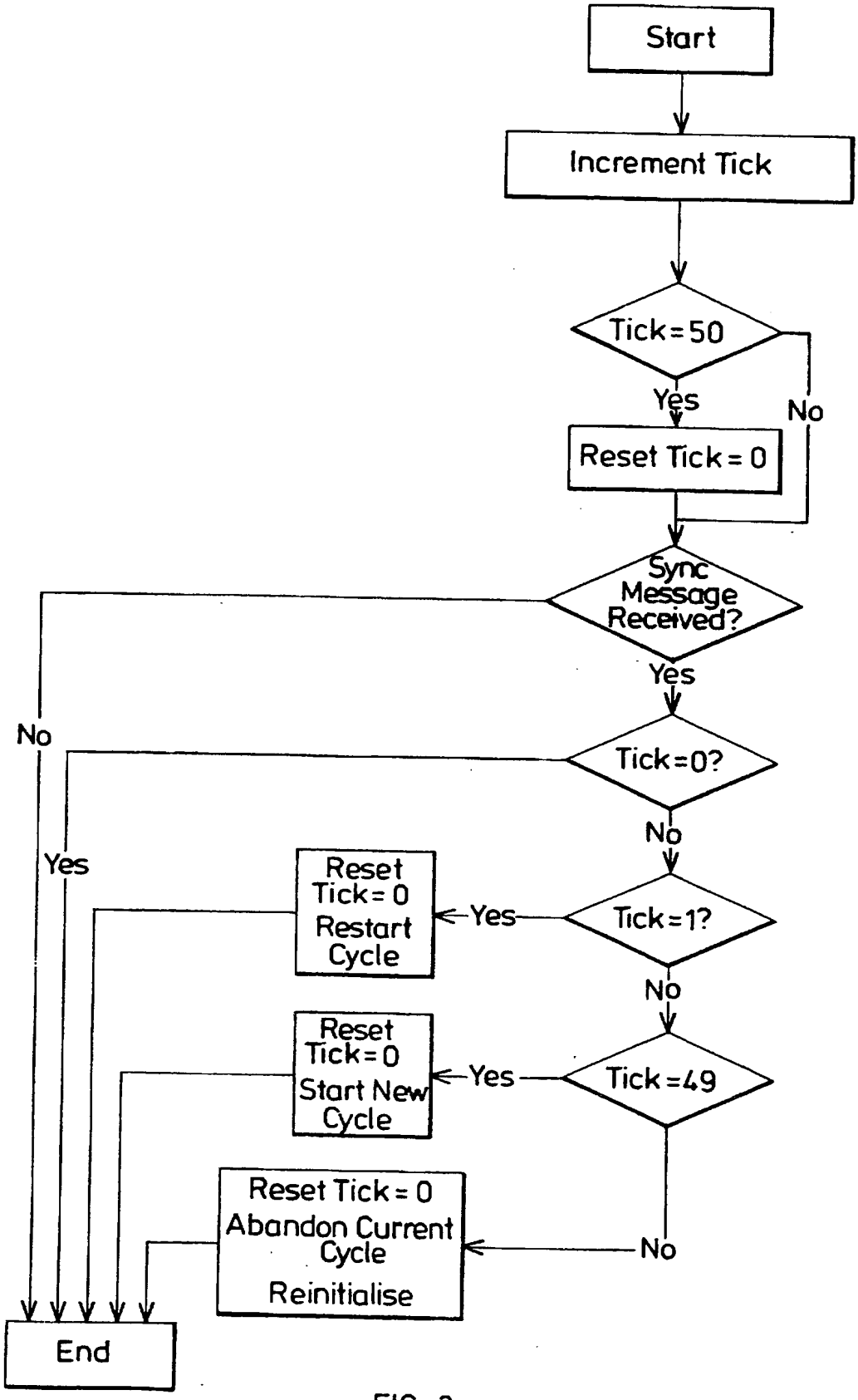

FIG 3

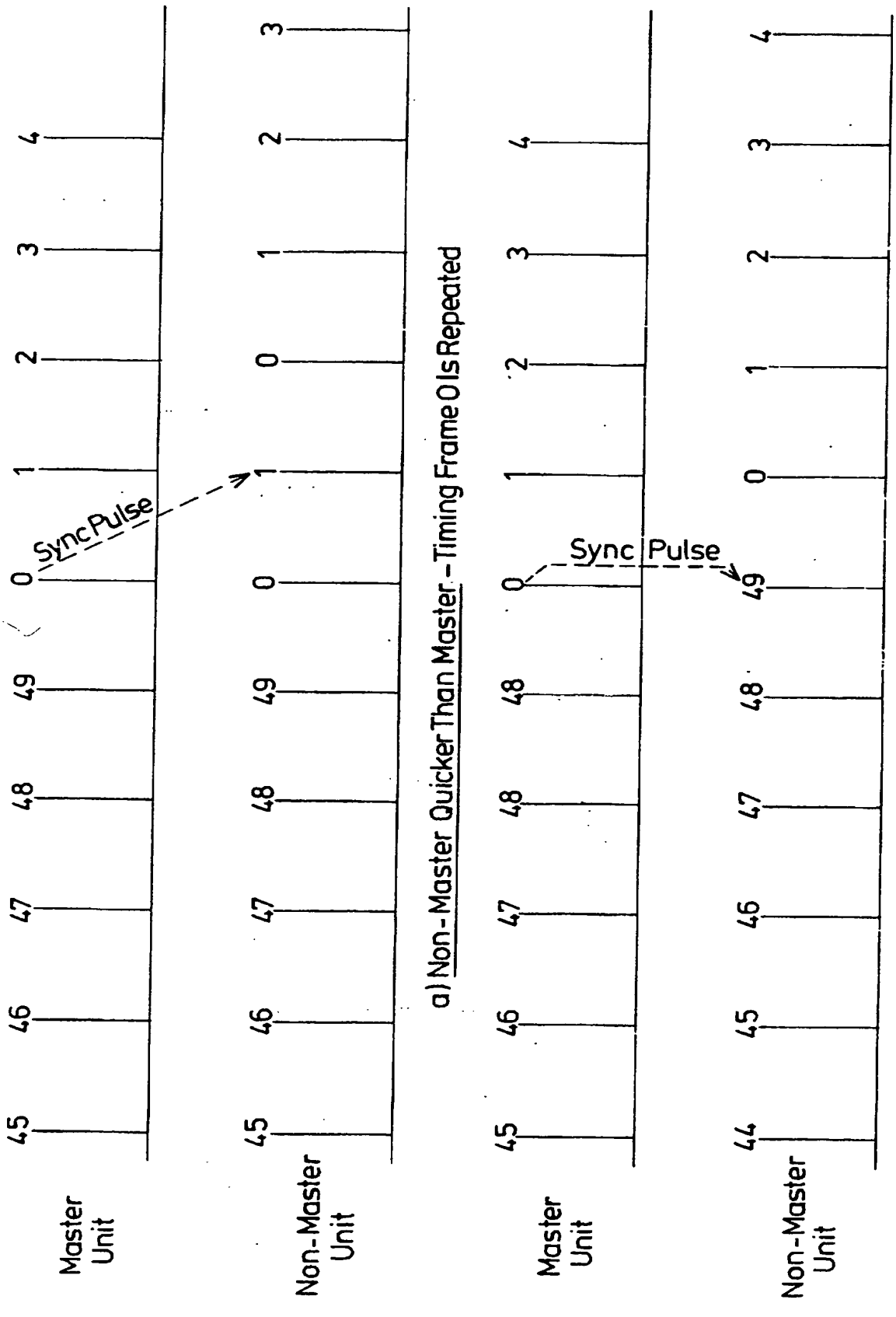

FIG 4

Distributed Data Processing System

This invention relates to a distributed data processing system which comprises a plurality of processing units at distributed locations. Each processing unit may have data acquisition means and/or controlled peripheral equipment and the units are linked by communication links so that data acquired by one unit can be processed thereby, and a processed result transmitted to another unit which provides appropriate output to a peripheral device.

By way of example, an aircraft engine control system may include three units. A first unit, referred to hereinafter as the fan-case unit, includes various transducers which sense engine operating input variables, such as air pressures and/or airflow rates and also receives inputs from cockpit controls either by communication links or by mechanical linkages into position transducers. A second unit, referred as the core unit, has data acquisition means which collects data from the vicinity of the engine core, such as the rotation speeds of various parts of the engine and internal air pressure data, and processes the locally collected data and that passed to it from the fancase unit to generate a fuel demand signal. The third unit is a fuel control unit which controls a fuel valve or valve/pump combination and provides a fast closed loop control of fuel flow to the engine in accordance with the fuel demand signal passed to it by the core unit. The three units have independent clock generators and are linked together by a serial bus system, which permits any number of units to broadcast and receive data. Various known bus systems which operate under protocols which enable bus contention problems to be solved are applicable.

In such a system, it is desirable to limit as far as possible the delays which occur between data processed by one unit being received by another unit

for processing. These delays do not occur in a centralised processing system for carrying out the same processes and should therefore be avoided. A possible method of reducing the delays would be to have the processor in each unit triggered into operation by an interrupt whenever there is data in a receive buffer awaiting processing. This is undesirable in any safety critical application, however, since the resulting system would operate in a non-deterministic manner, ie it would not be possible to predict the order in which control events occur.

Alternatively, tight synchronisation between the units could be employed, but this would involve dedicated hardware resources for synchronising the units with some added master clock.

It is an object of the present invention to provide a distributed processing system in which propagation delays between the units of the system is reduced to an acceptable level and the problems noted above are avoided.

In accordance with the invention there is provided a distributed data processing system comprising a plurality of independent cyclically operating processor units, a data communication system interconnecting said units, one of said units operating as a synchronisation master unit, which periodically transmits synchronisation messages to the other units via the communication system and the or each other unit operating to receive such synchronisation messages, to compare the timing of such messages with its own independent time frame and to shift its time frame dependent on the result of such comparison.

Each of the units preferably maintains its time frame by keeping a cyclic count of internally generated ticks, the cycle duration for each unit being nominally the same. The master synchronisation unit transmits a

synchronisation message at the end of each of its cycles (as its tick counter resets to zero). The or each other unit compares its tick count with a predetermined value each time it receives a synchronisation message. If the tick count is one less than the predetermined value, indicating that said other unit is lagging behind the master unit, the time frame of said other unit is shifted by starting a new cycle immediately. If the tick count is one, indicating that said other unit is leading the master unit by a whole tick interval, the time frame is shifted by restarting the cycle which has just started.

With such an arrangement, the or each other unit is maintained in loose synchronisation with the master unit such that it can be guaranteed to commence a processing operation following receipt of data, after a delay no longer than one tick interval.

In the accompanying drawings:

Figure 1 is a block diagram of an aircraft engine control system embodying an example of the invention;

Figure 2 is a flow chart of a synchronisation process carried out by one of the units of the system shown in Figure 1;

Figure 3 is a flow chart of a synchronisation process carried out on each of the other units; and

Figure 4 is a timing diagram showing the operation of the loose synchronisation arrangement.

Referring firstly to Figure 1, the distributed processor engine control

system shown comprises three separate units 10, 11, 12. The unit 10 is a fancase unit, i.e. it is installed on the engine structure adjacent the main air inlets. It has input connections from several transducers which provide signals representing various engine operating variables, such as air pressures and temperatures and air flow rates at various points around the air intake. The fan-case unit also receives input signals from cockpit transducers sensitive, for example, to the position of a pilot=s throttle lever. The second unit 11 is a core unit, which is installed in the structure of the engine close to the core thereof. This has inputs from further transducers which sense other engine parameters such as temperatures and pressures at the core of the engine and also the speeds of various independent shafts of the engine. The third unit 12 is a fuel control unit the primary function of which is to provide a fast closed loop control for a fuel control valve (or pump/valve combination).

The three units operate autonomously and they are connected together by a data communication system. In the example described a CAN (Control Area Network) bus is used and each of the three units 10, 11 and 12 includes a CAN Interface to enable the local host processor to transmit data to the bus and receive data therefrom. Each of the units has, for the purposes of description of the present invention, a similar architecture, comprising the CAN interface 13, a local host processor 14, a local clock generator 15 and a tick count register 16. Note that the tick count register is not a distinct hardware entity, but a location within the memory of the local host processor.

The units 10 and 11 operate separately to collect input data from their various transducer inputs and process it for use by the fuel control unit 12 in running its fast fuel control loop. Each of the units has a data processing cycle of roughly the same duration, so that the processing load can be

evenly spread between the CPUs of the three units. The CAN bus system used to interconnect the units ensures that bus contention problems are avoided, so that error free data transmission between the units can be achieved even in relatively noisy conditions.

The tick count registers 16 of the three units are used to enable loose synchronisation of the units to be achieved without added hardware being necessary. For the purpose of synchronisation, the fuel control unit 12 acts as a synchronisation master unit. As shown in Figure 2, a routine is included in the program of the CPU 14 of the unit 12 which executes every 2 mS and increments the count held in the register 12 until the count held in the register 16 reaches 50 (a 100mS period representing the maximum length of a processing cycle for any of the units). The CPU 14 of unit 12 then broadcasts a synchronisation message on to the bus and the register is reset to zero.

In each of the other units 10 and 11, the program routine shown in Figure 3 is executed every 2 mS. Each time the register 16 of the unit is incremented, a check is made to see if a synchronisation message has been received since the previous execution. If no message has been received, no action is taken. If a synchronisation message has been received the program checks to see if the tick count held in the register is zero, indicating that the register has just been reset and that the unit is in the required loose synchronism with the master unit 12. In this case no action is taken. If the tick count is not zero, the program checks whether its value is 1, which indicates that the unit is running faster than the master unit and is leading the master unit by more than 2 ms. If this condition occurs, the operating cycle of the unit which has just started is restarted and the tick count is reset to zero. If the value of the tick count is not one, the program checks to see whether the tick count is equal to 49, ie one

less than the expected number of ticks in an operating cycle. This indicates that the unit is running more slowly than the master unit and is lagging it. If this condition occurs, a new cycle is commenced immediately and the tick count is reset to zero. To ensure that important data is not lost as a result of the last part of the cycle being skipped in this way, the cycle of the unit should be designed so that only non-critical operations take place in the last sub-cycle. If the tick count has a value other than 1 or 49, the unit is not operating in loose synchronism with the master unit. In this case, therefore, the current processing cycle of the unit is abandoned, the process variables are all re-initialised and the tick counter is reset to zero.

Special provisions are made for circumstances in which a unit fails to synchronise with the master unit within a predetermined number of cycles or when no validated synchronisation messages have been received for a predetermined time-out period. The system is then put into a predetermined fail-safe mode.

With the arrangement described above, the fuel control unit 12 will always be able to access fresh data expected to be received from either of the other units 10, 11 with a delay of less than 2 ms. The units 10, 11 are not allowed to lag behind the master unit by any significant amount and there is a maximum of 2 ms between the sending of the fresh data from either of the units 10, 11 and the fuel control unit 12 being scheduled to access such data.

CLAIMS

1. A distributed data processing system comprising a plurality of independent cyclically operating processor units, a data communication system interconnecting said units, wherein one of said units is configured to operate as a synchronisation master unit, which periodically transmits synchronisation messages to the other units via the communication system, and the or each other unit is configured to operate to receive such synchronisation messages to compare the timing of such messages with its own independent time frame and to shift its time frame dependent on the result of such comparison.

2. A distributed data processing system according to claim 1, wherein each one of the cyclically operating processor units comprises means for maintaining its time frame by keeping a count of internally generated ticks, the cycle duration for each one of the units being nominally the same.

3. A distributed data processing system according to claim 2, wherein the master synchronisation unit comprises means for transmitting a synchronisation message to the or each other unit at the end of each of its cycles.

4. A distributed data processing system according to claim 3, wherein the or each other unit comprises means for determining whether the unit is lagging or leading the master unit by comparing its tick count with a predetermined value each time it receives a synchronisation message, the comparison indicating that said other unit is lagging if the tick count is one less than the predetermined value and indicating that said other unit is leading by a whole tick interval if the tick count is one,

and means for shifting the time frame of said other unit by starting a new cycle immediately when said other unit is determined to be lagging and for shifting the time frame by restarting the cycle which has just started when said other unit is determined to be leading.

5. A distributed data processing system according to claim 4, wherein the or each other unit has a processing cycle designed so that non-critical operations take place during the last sub-cycle to ensure that, in the event the determining means determines the unit to be lagging, important data is not lost as a result of the last part of the cycle being skipped.

6. A distributed data processing system according to claim 4, wherein the or each other unit has a processing cycle which is re-initialised and the current processing cycle abandoned if the comparison indicates that the tick count is more than one greater or less than the predetermined value.

7. An aircraft engine control system comprising a distributed data processing system according to any one of the preceding claims.

8. An aircraft engine control system according to claim 7, wherein the synchronisation master unit is a fuel control unit for providing a fast closed loop control for a fuel valve or pump/valve combination, and the other units include a fan-case unit for receiving signals representing engine operating input signals from cockpit transducers and a core unit installed in the structure of the engine close to the core thereof for receiving inputs from further transducers which sense other engine operating parameters.

9. An aircraft engine control system substantially as hereinbefore

described with reference to the accompanying drawings.

10. A distributed data processing system substantially as hereinbefore described with reference to the accompanying drawings.

The Patent Office

INVESTOR IN PEOPLE

Application No:	GB 9903612.1	**Examiner:**	Julyan Elbro
Claims searched:	1 to 10	**Date of search:**	8 September 1999

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.Q): G4A (AFT)

Int Cl (Ed.6): G06F 1/04, 1/06, 1/12, 9/46

Other: ONLINE: EPODOC JAPIO WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP 0438021 A2 IBM	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.